/\*

You are given a string, s, and a list of words, words, that are all of the same length.

Find all starting indices of substring(s) in s that is a concatenation of each word in words exactly once and without any intervening characters.

For example, given:

s: "barfoothefoobarman"

words: ["foo", "bar"]

You should return the indices: [0,9].

用map存所有word，然后每次从s中取alllength长度出来，然后找word里面有的话，就--

如果已经为0了，就删除。

\*/

class Solution {

public:

vector<int> findSubstring(string s, vector<string>& words)

{

vector<int> result;

int wlength=words[0].size();

//判断是否超长

int alllength=wlength\*words.size();

if(s.size()<alllength)

return result;

//存进map里面

map<string,int> m1;

for(int i=0;i<words.size();i++)

m1[words[i]]++;

string temp;

int k;

map<string,int> mm;

map<string,int>::iterator it;

for(int i=0;i<=s.size()-alllength;i++)

{

mm=m1;

k=i;

while(mm.size()!=0)

{

it=mm.find(s.substr(k,wlength));

if(it==mm.end())

break;

else

{

mm[s.substr(k,wlength)]--;

if(mm[s.substr(k,wlength)]==0)

mm.erase(it);

}

k+=wlength;

}

if(mm.size()==0)

result.push\_back(i);

}

return result;

}

};